

Examen

105000016 - Programación para Sistemas Grado en Ingeniería Informática (2009)

Lenguajes y Sistemas Informáticos e Ingeniería de Software

Facultad de Informática

Universidad Politécnica de Madrid

Curso 2014/2015 - Junio 2015

Normas

- El examen puntúa sobre **8 puntos**.
- La duración total del mismo es de **35 minutos**.
- Se deberá tener el DNI o el carnet de la UPM en lugar visible.
- No olvidar rellenar **apellidos, nombre y número de matrícula** en cada hoja.
- La solución al examen se proporcionará antes de la revisión.

Cuestionario

- (1 punto) 1. Se pide escribir el comando necesario para compilar el código que se ha escrito en el fichero principal.c, si dicho fichero utiliza la librería auxiliar.h vista en clase. El comando pedido debe generar un fichero ejecutable llamado: ejecutable_examen

Solución: gcc principal.c auxiliar.c -o ejecutable_examen
--

- (1 punto) 2. Dado el siguiente código C

```
#include <stdio.h>
main() {
    int vector [10];
    int *ptr= vector;
    int cont=0;
    *ptr = 10;
    for (cont=1; cont<10; cont++) {
        *(ptr+cont)=*(ptr+cont-1)+5;
    }
}
```

```
    for (cont=0; cont<10; cont=cont+2) {  
        printf(" %d: %d-", cont, *(ptr+cont));  
    }  
}
```

Indicar cual será la salida por pantalla al ejecutar dicho código.

Solución: 0:10-2:20-4:30-6:40-8:50-

(1 punto) 3. Dado el siguiente código C

```
#include <stdio.h>  
int valor = 45;  
void incrementar(int *valor);  
main() {  
    int valor = 22;  
    incrementar(&valor);  
    printf(" %d_", valor++);  
    incrementar(&valor);  
    printf(" %d_", --valor);  
}  
void incrementar(int *num) {  
    int valor = 37;  
    *num = 12;  
}
```

Indicar cual será la salida por pantalla al ejecutar dicho código.

Solución: 12 11

(1 punto) 4. Dado el siguiente fragmento de código:

```
#include <stdio.h>  
#include <stdlib.h>  
main(int argc, char *argv[]) {  
    FILE *fp;  
    if (argc<3) {  
        printf ("Uso: _programa_nombre_fichero_texto\n");  
        exit (1);  
    }  
    ... (1) ...  
    if (fp==NULL) {  
        printf ("Fichero_no_existe");  
        exit (1);  
    }  
    ... (2) ...  
    fclose ( fp );  
    return 0;  
}
```

Escribir las 2 líneas de código faltante para que el programa escriba en un fichero, pasado como primer argumento al programa, el texto pasado como segundo argumento. El fichero debe truncarse para escribir el nuevo texto.

Solución: ...**(1)**... `fp = fopen (argv[1], "w");`
...**(2)**... `fputs (argv[2],fp);`

(1 punto) 5. Dada el siguiente fragmento de código:

```
typedef struct {  
    int dia, mes, anyo;  
} fecha;
```

```
fecha hoy = {15,6,2015}, ayer;  
ayer = hoy;
```

Escribir una única sentencia que permita almacenar en la variable *ayer* los valores {14,6,2015} utilizando el operador decremento `--`.

Solución: `ayer.dia--;`

(1 punto) 6. Completar el siguiente código para que permita crear un vector de números enteros cuya longitud es tecleada por el usuario incluyendo la comprobación de que se ha podido crear el vector necesario.

```
include <stdio.h>  
#include <stdlib.h>  
  
int main() {  
    int n,i,*ptr;  
    printf("Longitud_del_vector:_");  
    scanf("%d",&n);  
}
```

Solución:

```
ptr=(int*)malloc(n*sizeof(int));  
    if (ptr==NULL)  
    {  
        printf("Error_en_la_petición_de_memoria");  
        exit(-1);  
    }
```

(1 punto) 7. Una vez iniciada la depuración de un programa, indique los comandos que ejecutaría en el depurador gdb para detener la ejecución cuando la variable *numero* cambie de valor y posteriormente continuar con la ejecución del programa

Solución: watch numero
continue

(1 punto) 8. Dado el siguiente código, indique la salida que produce:

```
#include<stdio.h>

struct ejemplo{
    int a;
    char ch;
};

int main(void)
{
    struct ejemplo objeto={1,'a'};
    struct ejemplo *objeto2 = &objeto;

    objeto2->a = 2;
    objeto2->ch = 'b';

    printf("%d_ %c\n",objeto.a, objeto.ch);
    return 0;
}
```

Solución: 2 b